Client Reference No. P01-1760

APPLICATION FOR UNITED STATES LETTERS PATENT

For

ON THE FLY SUMMARIZATION OF FILE WALK DATA

Inventors:

Vijay Deshmukh

Benjamin Swartzlander

Timothy J. Thompson

Prepared by:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
32400 Wilshire Boulevard
Los Angeles, CA 90025-1026
(408) 720-8300

Attorney's Docket No.: 005693.P047

"Express Mail" mailing label number: EV409365137US
Date of Deposit: March 12, 2024
I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria VA 22313-1450

Patricia Richard
(Typed or printed name of person mailing paper or fee)

(Signature of person mailing paper or fee)

3/12/2024
(Date signed)

On the Fly Summarization of File Walk Data

FIELD OF THE INVENTION

[0001] At least one embodiment of the present invention pertains to networked storage systems, and more particularly to a method and apparatus for collecting and reporting data pertaining to files stored on a storage server.

BACKGROUND

[0002] A file server is a type of storage server which operates on behalf of one or more clients to store and manage shared files in a set of mass storage devices, such as magnetic or optical storage based disks. The mass storage devices are typically organized as one or more groups of Redundant Array of Independent (or Inexpensive) Disks (RAID). One configuration in which file servers can be used is a network attached storage (NAS) configuration. In a NAS configuration, a file server can be implemented in the form of an appliance, called a filer, that attaches to a network, such as a local area network (LAN) or a corporate intranet. An example of such an appliance is any of the NetApp Filer products made by Network Appliance, Inc. in Sunnyvale, California.

[0003] A filer may be connected to a network, and may serve as a storage device for several users, or clients, of the network. For example, the filer may store user directories and files for a corporate or other network, such as a LAN or a wide area network (WAN). Users of the network can be assigned an individual directory in which they can store personal files. A user's directory can then be accessed from computers connected to the network.

[0004] A system administrator can maintain the filer, ensuring that the filer continues to have adequate free space, that certain users are not monopolizing storage on the filer, etc. A Multi-Appliance Management Application (MMA) can be used to monitor the storage on the filer. An example of such an MMA is the Data Fabric Monitor (DFM) products made by Network Appliance, Inc. in Sunnyvale, California. The MMA may provide a Graphical User Interface (GUI) that allows the administrator to more easily observe the condition of the filer.

[0005] The MMA needs to collect information about files stored on the filer to report back to the administrator. This typically involves a scan, also referred to as a "file walk," of storage on the filer. During the file walk, the MMA can determine characteristics of files stored on the filer, as well as a basic structure, or directory tree, of the directories stored thereon. These results can be accumulated, sorted, and stored in a database, where the administrator can later access them. The MMA may also summarize the results of the file walk so they are more easily readable and understood by the administrator.

[0006] On a filer that manages a large amount of storage, the file walk can be a very resource intensive process. The results of a file walk typically must be processed after the walk is completed so that the results are easy for an administrator to comprehend. An MMA typically has many tasks to perform, and generally should be available for the administrator. What is needed is a way to reduce the load on an MMA while still maintaining and monitoring attached appliances.

SUMMARY OF THE INVENTION

[0007] A method for collecting data from a storage server is disclosed. A directory on the storage server is scanned. The number of children in the directory is determined, and the number is added to a reference count. The child is scanned to collect information about the child, and the information is combined into a summary of the directory. The reference count is reduced after scanning the child.

[0008] Other aspects of the invention will be apparent from the accompanying figures and from the detailed description which follows.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] One or more embodiments of the present invention are illustrated by way of example and not limitation in the figures of the accompanying drawings, in which like references indicate similar elements and in which:

[0010] **Figure 1** illustrates a monitoring system for a storage server;

[0011] **Figure 2** illustrates a block diagram of an agent;

[0012] **Figure 3** is a flowchart illustrating a process for pre summarizing and analyzing results generated by an agent;

[0013] **Figure 4** illustrates a table displaying a list of interesting files

[0014] **Figure 5** illustrates a table listing information about directories on the server; and

[0015] **Figure 6** illustrates a histogram showing server usage of certain users.

DETAILED DESCRIPTION

[0016] Described herein are methods and apparatuses for On the Fly Summarization of File Walk Data. Note that in this description, references to "one embodiment" or "an embodiment" mean that the feature being referred to is included in at least one embodiment of the present invention. Further, separate references to "one embodiment" or "an embodiment" in this description do not necessarily refer to the same embodiment; however, such embodiments are also not mutually exclusive unless so stated, and except as will be readily apparent to those skilled in the art from the description. For example, a feature, structure, act, etc. described in one embodiment may also be included in other embodiments. Thus, the present invention can include a variety of combinations and/or integrations of the embodiments described herein.

[0017] According to an embodiment of the invention, an agent, which is a server separate from a storage server, scans the storage server to determine information about files stored on the server. The information may include statistics such as the location and size of files, the date of creation of files, etc. The storage server may be scanned by one or more threads. While the agent is scanning the server and collecting information about the files, the server is generating summaries and updating existing summaries. The data may be collected and summarized on the fly, and since no further processing is required, the resulting summary can immediately be loaded onto a database server once the file walk is completed. At that point, an administrator can query the summary without requiring any further processing.

[0018] The MMA is generally a single server that is used to allow a system administrator to monitor a storage or file server. When a large storage server is

monitored, the MMA may have difficulty performing its monitoring duties and a file walk at the same time. In fact, the file walk may make the MMA inaccessible to the system administrator, and the MMA may become a bottleneck, since it may be incapable of performing the file walk in a reasonable amount of time. According to an embodiment of the invention, independent agents are used to perform the file walk, to reduce the load on the MMA. The independent agents can run one or more threads to perform a file walk of the storage server. These threads can generate summaries about the files while the scan is occurring, so that less processing time is required when the summaries are queried at a later time.

[0019] **Figure 1** illustrates a monitoring system for a storage server. The system 100 includes a filer 102, an MMA 104 including a monitor 106, a database 108, a graphical user interface (GUI) 110, and two agents 112 and 114. The agents 112 and 114 can perform a file walk of the filer 102 for the MMA 104. An agent may be an independent server that is attached to the network and is dedicated to performing file walks. By having an agent perform this task rather than having the MMA do it, the MMA can save its resources for other tasks, such as monitoring current activity on the filer 102 using the monitor 106. Ultimately, one goal is to minimize the amount of work the MMA is required to do.

[0020] Multiple agents can be added to perform a complete file walk in less time if necessary. For example, the MMA 104 may instruct two or more agents to each scan a subset of the directories found on the filer 102. Using multiple agents ensures that the file walk will be completed in less time. A system administrator can, through the MMA

104, customize the file walk, so that any number of agents can be used to increase the speed of the file walk.

[0021] According to one embodiment of the invention, the agents 112 and 114 may use a file system different from the one used by the filer 102. For example, the agent 112 uses the Common Internet File System (CIFS), while the agent 114 uses the Network File System (NFS). Here, either agent 112 or 114 is able to perform the file walk of the filer 102, regardless of the file system used by the filer 102. The agent 112 also has storage 116 to store the results of a file walk while the walk is occurring and before they are transferred to the MMA 104. The agent 114 may also have attached storage for this purpose.

[0022] The filer 102 is generally attached to a volume 118. The volume 118 may include one or more physical hard drives or removable storage drives that comprise the storage for the filer 102. For example, the volume 118 may comprise a RAID structure. The filer 102 may also be connected to other volumes that comprise storage. A file walk generally scans all files stored on the entire volume 118, regardless of whether all of the files are stored on the same physical drive. Further, although the volume 118 may contain several separate physical drives, the volume 118 may appear and function as a single entity.

[0023] The results of a file walk may be transferred to and stored on the database server 108 after the file walk is complete. The database server 108 can then be accessed by the GUI 110, so that an administrator can search the results of the file walk. The GUI 110 may allow the administrator to easily parse the results of a specific file walk, including allowing the administrator to monitor the total size of files stored on the filer,

the size of particular directories and their subdirectories, the parents of specific directories, etc. These queries will be discussed in more detail below. The file walk may also collect statistics about the files on the filer, such as the total size of files, the most accessed files, the types of files being stored, etc. According to one embodiment, the GUI 110 may be a web-based Java application.

[0024] According to an embodiment of the invention, the summary is written to the database server 108 as a table or a histogram. The summary may then be accessed through a Java applet using a web browser such as Internet Explorer or Netscape. In another embodiment, the summaries are accessed using other programs. Although tables are shown here, it is understood that any appropriate manner of relaying the summary data to the administrator may be used.

[0025] **Figure 2** illustrates a block diagram of an agent. The agent 112 includes a processor 202, a memory 204, a network adapter 206, and a storage adapter 208. These components are linked through a bus 210. The agent 112, as shown in **Figure 2**, is typical of a network server or appliance, and it is understood that various different configurations may be used in its place. The agent 114 may be similar.

[0026] The processor 202 may be any appropriate microprocessor or central processing unit (CPU), such as those manufactured by Intel or Motorola. The memory 204 may include a main random access memory (RAM), as well as other memories including read only memories (ROM), flash memories, etc. The operating system 212 is stored in the memory 212 while the agent 112 is operating. The operating system includes the file system, and may be any operating system, such as a Unix or Windows based system. The network adapter 206 allows the agent 112 to communicate with

remote computers over the network 214. Here, the agent 112 will be collecting data from the filer 102 and sending data to the MMA 104. The storage adapter 208 allows the agent 112 to communicate with the storage 116 and other external storage.

[0027] **Figure 3A** illustrates a directory tree. The directory tree 300 may show a relationship between directories stored on a volume such as the volume 118. Each of the nodes 301-310 symbolizes a directory. The first node 301 is a "parent" to all other directories. Likewise, all the other directories are "children," or "child nodes" of the node 301. As another example, the node 303 is an "immediate child" of the node 302, and the node 302 is the parent of the node 303. The nodes 303 and 306 are "siblings," they both have the same parent and are located on the same level of the tree 300. The tree 300 can be used to represent a relationship between directories stored on a volume. The nodes 301-310 may be assigned identification (ID) numbers. The ID numbers shown here are 1-10. According to an embodiment of the invention, the ID numbers are assigned in a depth first search (DFS) order, in which the ID numbers are first assigned down through the tree, and then across the tree. The ID numbers can be used to identify specific directories.

[0028] **Figure 3B** illustrates the relationship between the directories in the tree 300. The structure 350 shows a directory structure of a typical volume 118. The directory structure 350 corresponds to the tree 300. The directory structure 350 shows how the directories are embedded within one another.

[0029] **Figure 4** is a flowchart illustrating a method of writing summarized data on the fly. During the file walk scanning, the agent 112 collects information about files stored on the volume 118. This information may include file names, directory names, file

sizes, dates of creation, etc. The file walk may be performed by one or more 'threads.' A thread may be a program capable of operating independently of other programs. Using a single threaded system, the agent scans directories and files found on the volume 118 with a single thread. A multi-threaded system may include two or more threads. According to one embodiment, a file thread can be used to scan and determine characteristics of files, while a directory thread can be used to determine the contents of directories. A directory queue and a file queue are also established. The directory queue contains a list of directories to be examined. Likewise, the file queue contains a list of files to be examined.

[0030] The directory thread examines the directory found at the top of the directory queue, and places that directory's contents into the file queue. The file thread then examines the members of the file queue, placing directories in the directory queue and examining files. The file thread may collect information including the name of the file, the size of the file, the location of the file, the type of file, the time of creation of the file, the time of last access of the file, and the owner of the file. This information will be used to create tables and histograms such as the table in **Figure 5** and the histogram in **Figure 6**. While the information is collected, the table and summaries are created on the fly. In other words, the agent 112 combines information collected into already established summaries as soon as the information is collected. The directory thread may also report information about the directory structure on the volume 118.

[0031] In block 402, a directory is scanned. The directory may be scanned by the directory thread, as mentioned above. The directory will typically be removed from the top of the directory queue. As an example, the directory symbolized by the node 303 is

first scanned here. In block 404, a number of children or child nodes in the directory is determined. As can be seen in the tree 300, the node 303 has two children that are directories. A directory's children may also be the files stored within that directory. As an example, say that there are two files stored in the directory. Therefore, the directory '/u/employees/a-m' has four children. In block 406, the number of children is added to a reference count for the directory. Here, the reference count will have four added to it. If the directory thread had just begun scanning the directory, the reference count will start at zero. So here, the reference count will be four.

[0032] In block 408, a child of the directory is scanned. The child nodes of the directory may be listed in the file queue when the directory is scanned in block 402. The children of the directory may either be files or directories. The file thread determines whether the children are files or directories. If the child is a directory, the child is added to the front of the directory queue. If the child is a file, it is analyzed, its characteristics are determined, and combined with the characteristics of its parent in block 410. The characteristics of a directory are cumulatively summarized. For example, the agent 112 may keep a running total of the number of files found within a directory. If a file is encountered, this number is added to the total for either the current directory or for a parent of the current directory. In one embodiment, cumulative totals are kept for all directories. In other words, the statistics for a specific directory include the file statistics for *all* of the directory's children. For example, the cumulative total for the directory represented by the node 303 would include not only the files found within that directory, but also the files found within the directories represented by the nodes 304 and 305.

[0033] The agent 112 may keep track of many statistics. These statistics are updated during the file walk. For example, the largest and smallest file found may be tracked. When scanning a file, the agent 112 compares the size of the file with the size of the current largest and smallest file found. If the current file is either larger than the largest or smaller than the smallest file found, the current file becomes the new largest or smallest file. Other statistics are also tracked on the fly. For example, averages are reweighted with the statistics of the new file. Also, histogram counters are summed, a file's size is added to the total for the directory, the owner of the file is tracked and used for histograms, etc.

[0034] In block 412, the reference count is reduced by 1. This way, the file thread will know when no more children need to be scanned. Since the reference count will be equal to the number of children of a directory, the reference count ensures that all of the children of the directory are examined. In block 414, if the reference count is zero, the process 400 is finished. If the reference count is greater than zero, the process returns to block 408, where another child is examined.

[0035] The root directory is the parent of all other directories found on the filer 102. When the file walk of the root directory is completed, or when the root directory's reference count equals zero, the entire file walk is completed. As a result, the summaries and histograms are also completed, and may be imported into the database on the database server 108. According to an embodiment of the invention, the summaries and histograms are stored in a format compatible with the database software used on the database server 108. For example, the database server 108 may be running a database program by Sybase, Inc. The Sybase database includes a "LOAD TABLE" command.

In order to expedite the transfer of the summaries and other data onto the database server 108, the data may be stored in a format that is compatible with the "LOAD" command. It is understood that other database suppliers may use different commands, and that the data may also be formatted to be used with these other commands.

[0036] The process 400 describes creating summaries of information about files stored on a volume 118. These summaries are created and updated while the agent 112 is initially examining the files, rather than at a later time. By having the agent 112 update and create summaries during the examination process, the process is simplified, and the amount of processing required is reduced.

[0037] **Figures 5 and 6** show graphical representations of the data stored on the server. **Figure 5** illustrates a table listing information about directories stored on the filer 102, and **Figure 6** shows a histogram displaying usages by certain users. These graphical representations may be created during the file walk using the process 400. These graphical representations are examples of representations of data. It is understood that many other representations are possible.

[0038] **Figure 5** illustrates a table 500 listing information about directories on the server 102. The table 500 is an example of information and data that may be reported to an administrator. This data is useful for maintaining the filer 102. The table 500 includes several columns, listing the directory name in the column 502, the number of files in the directory in the column 504, the total size of the files in the directory in the column 506, and the average time of the last access to files in the directory in the column 508. The agent 112 collects this information during the file walk, and compiles the table. The MMA 104, in many instances, does not have the resources to generate these tables or

collect these results. This is especially true where there are several agents scanning a single storage server. Having the agents perform these tasks will save resources that the MMA 104 may require for other tasks.

[0039] The collected information about the directories on a storage server can be useful for several reasons. The administrator can find bottlenecks in the system, as well as directories that have an abnormally large number of files or total size. In other embodiments, another table, similar to the table 500 may be generated. This table may include cumulative statistics that list the total number of files in a directory, including the total statistics for all embedded directories found within that directory. For example, the column 506 may list the total size of all files in a directory and in the directory's subdirectories.

[0040] The column 508 lists the average last access time for the files located in the listed directory. The column 508 lists a time stamp, in other words, an average time during which all files in the directory were last accessed. For example, if a directory contained five files, one most recently accessed today, one yesterday, one two days ago, one three days ago, and the last four days ago, the average access time would be sometime two days ago. This is useful so that an administrator can easily determine how active the particular directory is, and whether there are a large number of files that are not being regularly accessed. For example, it appears that there are a number of stale files in the directory '/u/users/a/aaron/' since the average access time is over eighteen months ago.

[0041] **Figure 6** illustrates a histogram showing server usage of certain users. The histogram 600 demonstrates how much space each user is occupying on the volume 118.

An administrator can use this data to determine whether one user is occupying an abnormally large amount of space. In one embodiment, the MMA 104 can use this information to revoke the user's ability to store any more files. For example, the users 'Aaron' and 'Gibson' are using much more storage space than the other users. The administrator can target these users to increase the amount of free space on the server, if needed.

[0042] The histogram 600 may be personalized by the administrator. For example, in a system with many users, it may be difficult for the administrator to parse the histogram 600. Therefore, the histogram 600 may list the users with the highest usage first, or only those users that are using more than a specified amount of storage space. A histogram showing the usage of many users may allow an administrator to determine the approximate percentage of users that are using an abnormally large amount of server space. It is understood that the data represented in the histogram 600 may also be displayed in other forms, such as in table form.

[0043] Although not shown, the histogram 600 may also show a number of files, in addition to, or in place of the size of the files as shown. For example, the agent 112 may generate a second histogram that shows the number of files stored by a certain user or the number of files of a certain file type. Further, the histogram 600 may include a second field for each user or file type that lists the number of files. It is also understood that other useful information may be displayed in this manner.

[0044] The techniques introduced above have been described in the context of a NAS environment. However, these techniques can also be applied in various other contexts. For example, the techniques introduced above can be applied in a storage area network

(SAN) environment. A SAN is a highly efficient network of interconnected, shared storage devices. One difference between NAS and SAN is that in a SAN, the storage server (which may be an appliance) provides a remote host with block-level access to stored data, whereas in a NAS configuration, the storage server provides clients with file-level access to stored data. Thus, the techniques introduced above are not limited to use in a file server or in a NAS environment.

[0045] This invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident to persons having the benefit of this disclosure that various modifications and changes may be made to these embodiments without departing from the broader spirit and scope of the invention. The specification and drawings are accordingly to be regarded in an illustrative, rather than in a restrictive sense.